REMARKS

Pursuant to the Examiner's request for a copy of Applicant's response on floppy disk, there is attached hereto a 3½" IBM format floppy disk with an electronic file copy in WordPerfect.

Claims 1 to 9 remain in the application. Claim 9 has been amended to correct a minor error resulting from the amendment filed on December 21, 2004.

Claims 1 to 9 were rejected under obviousness-type double patenting as being unpatentable over claims 1 to 9 of co-pending application 09/639,931 filed August 17, 2000, and issued as U.S. Patent No. 6,683,624. In making the rejection, the Examiner noted that a timely filed terminal disclaimer in compliance with 37 C.F.R. 1.321(b) would overcome the rejection. Submitted herewith is a terminal disclaimer, both this application and U.S. Patent No. 6,683,624 being commonly owned by International Business Machines Corp. Withdrawal of the rejection under obviousness-type double patenting is therefore respectfully requested.

Claims 1 to 9 were additionally rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 5,883,639 to Walton et al. in view of U.S. Patent No. 6,446,081 to Preston. This rejection is respectfully traversed for the reason that the combination of Walton et al. and Preston does not fairly teach or suggest the claimed invention.

The claimed invention is directed to a visual programming language in which programming objects in memory are represented as graphical objects on a display and wherein when program properties of the programming objects change, changes are induced in the graphical properties of the visual representation of the object on the display. The invention is used in static program analysis which aims at determining properties of the behavior of a program without actually executing it. Static analysis is founded on the theory of abstract interpretation for proving the correctness of analyses with respect to the semantics of a programming language.

In the Office Action mailed September 21, 2004, the Examiner had rejected claims 1 to 9 under 35 U.S.C. §102(e) as being anticipated by the patent to Walton et al. The amendment filed with the Request for Continued

Examination filed December 21, 2004, amended claims 1, 8 and 9, and presented arguments distinguishing the claimed invention from Walton et al. In response to Applicant's arguments as made in that amendment, the Examiner states that the arguments "have been considered but are moot in view of the new grounds of rejection." However, in making this new ground of rejection, the Examiner has not fully taken into consideration those arguments which are herein incorporated by reference.

Walton et al. disclose a visual software engineering system which uses a concept of defining both input to and output from graphical objects in an object-oriented system by providing examples of what the user desires the graphical object to do. This technique is referred to by Walton et al. as "animation by example". In this process, the user creates a user interface by drawing the user interface with a graphics editor and then defining the output behavior (i.e., graphics manipulation) of the user interface components by showing each state or frame as an animation. Thus, Walton et al. disclose a visual software engineering system for providing a means of *creating user interfaces* to products under development *without the need of a programming language*. On a graphical interface, the user *designs* a user interface, including visual artifacts such as buttons, check boxes, etc. These visual entities are managed by "graphical objects". A graphical object is not a generic programming object, but rather is an object subject to enablement. Walton et al. state at column 8, lines 54–56,"The resulting objects are then stored as objects in an object-oriented database system and connected to other objects or user code . . ." The user code has to "connect to" or invoke the graphical objects. The graphical objects are not user code defined objects; rather, they are service objects to the user application program.

A distinguishing feature of the disclosed invention is found in the specification on page 9, beginning at line 16, wherein implicitly the programming objects (variables, objects) are part of *an existing or developing* application program. What this means is that the program property of the programming object is derived from some kind of code analysis. The claimed invention is directed to a mechanism for displaying a change in a program property of a programming element..

In contrast to Walton et al.,

1) The claimed programming objects refers to user application objects (variables, etc.), not the specific graphical objects (gauges, slide bars, buttons, etc.) as described in Walton et al;
2) The program property of the programming object is derived from a program context and the state is displayed. This is the opposite direction from Walton et al., who defines state and stores it, and maps actions. The former is a reflective paradigm, while the latter is more of a constructive paradigm.

The Examiner has failed to acknowledge these quite clear distinctions and instead persists in attempting to construe Walton et al. as if it were directed to the same type of system as claimed. The fact is that Walton et al. have an entirely different purpose, a graphical user interface design tool. There is no disclosure in Walton et al. about a visual programming language. There is no disclosure in Walton et al. of *programming objects* in memory being represented as graphical objects on a display and wherein when *program properties* of the *programming objects* change, changes are induced in the graphical properties of the visual representation of the object on the display. Whereas Walton et al is used to design a graphical user interface, the claimed invention is used in static program analysis which aims at determining properties of the behavior of a program without actually executing it.

The Examiner states that "Walton does not explicitly disclose the limitation of a graphical property representing a programming object's property." Applicant must object to this characterization. The use of the word "explicitly" would seem to imply that Walton et al. might some how "implicitly" disclose such a feature when, in fact, there is not even so much as a hint of it, as brought out in the foregoing analysis. The Examiner then goes on to cite Preston as teaching "the program properties of the programming objects are reflected through graphical elements (the shaped displayed is selected so as to match the recognized characteristics . . . to indicate whether it is an entity or a state of affairs, p19 48–51)." The Examiner's citation is incorrect. The complete quote appears on column 20, lines 42–46, of Preston as follows:

> "In such cases, the text held in the record of the server 200 is augmented by a pointer to the lexical database entry, and the shape displayed is selected so as to match the recognised characteristics of the text (for example, to indicate whether it is an entity or a state of affairs)."

Note that the Examiner has left out "of the text" following the word "characteristics". This description is part of the description of the sixth embodiment. The Examiner also states that "the 'visual display to represent the output data', p17 32–39 would enable users to more easily ascertain/comprehend the functional components requisite in the specified program." Again, the Examiner's citation is not correct. The complete quote is found at column 18, lines 11 to 20, as follows:

> "The applet 116 may also be arranged to generate a visual display to represent the output data; for example, a representation of a human face, or entire human head, animated in synchronism with the output speech as described in our earlier application EP-A-225729, or a sign language display comprising an animated representation of a pair of hands generating sign language (for example British or American sign language) from a text to sign language converter program. This latter embodiment is particularly advantageous for those with hearing difficulties."

This is part of the description of the fourth embodiment. Both the fourth and sixth embodiments of Preston are directed to data storage and retrieval, not a visual programming language.

Preston discloses an input apparatus for a data processing system in which a free-form source document is input and processed to parse a source document to locate semantically meaningful entities and to store corresponding content data. A graphical display is arranged to generate a visual representation of the source document in which the semantically meaningful entities are represented by pictorial elements. At column 2, lines 33–45, Preston states that his invention differs "from so called 'visual programming' systems . . . [which] provide a graphic environment in which operations to be specified are represented visually, and a user may specify a sequence of such operations by editing the display to create and later linkages between the elements." In the Preston data input and retrieval apparatus, semantic structures, corresponding to the graphical

representation (corrected where necessary), are stored for subsequent processing or retrieval. In one embodiment, shown in Figure 21 and described as the fifth embodiment at column 18, line 21, to column 19, line 2, the stored data is employed by a code generator, to generate a computer program. There is, however, nothing in this description that would suggest that graphical aspect changes in color, position and size of programming objects reflect changes in a program property of the programming objects, as specifically claimed.

Claim 1 recites a "computer implemented method of visual representation of programming objects as graphical elements, *wherein program properties of said programming objects are reflected through graphical properties of graphical elements*" (emphasis added). The method comprises three steps. The first step is "*detecting a change in a program property of a programming object* in visual representation and shown visually on a display device as one or more graphical elements, wherein graphical elements represent the programming object and *program properties of programming objects are reflected through graphical element properties*" (emphasis added). The second step is "*determining graphical aspect changes that apply to graphical elements of the programming object appropriate for the change in a program property of the programming object*" (emphasis added). The third step "applying the graphical aspect changes to corresponding graphical elements, wherein the graphical aspect changes include changes in color, position and size." This last step is the process of *state reflection* "wherein programming properties of programming objects are reflected through graphical properties of graphical elements", as recited in the preamble.

As explained above, Walton et al. do not contemplate the process of *state reflection* as a result of detecting a change in state of a data element representing a programming object, and there is nothing in Preston which would suggest this.

Claim 8 recites an "apparatus for visual representation of programming objects as graphical elements". This apparatus comprises "a data processing system comprising a display device, an interactive device, as in a keyboard, a pointing device, a storage device, and a data processor", a "memory coupled to the data processor via a bidirectional bus, wherein the memory includes a first memory section for at least one program and a second memory section for data",

"computer code . . .", and "means for displaying a visual representation of a plurality of graphical elements on the display device. . ." The computer code comprises "a visual programming language, wherein the computer code is stored in the first memory section, and the computer code *detects a change in a program property* of a programming object, *determines graphical aspect changes that apply to graphical elements* which represent the programming object, and *applies graphical aspect changes to said visual representation of said programming object which represents the change of the program property of the programming object*" (emphasis added). The "displayed graphical elements represent programming objects and program properties of programming objects are reflected through displayed graphical element properties." Thus, claim 8 similarly defines over the combination of Walton et al. and Preston.

Claim 9 is similar to claim 1, but is directed to a machine readable medium containing code for performing the method of claim 1. The language of claim 9 is otherwise similar to that of claim 1.

The Examiner is reminded of the basic considerations which apply to obviousness rejections as set out in MPEP 2141. Specifically, "When applying 35 U.S.C. 103, the following tenets of patent law must be adhered to:

"(A) The claimed invention must be considered as a whole;

"(B) The references must be considered as a whole and must suggest the desirability and thus the obviousness of making the combination;

"(C) The references must be viewed without the benefit of impermissible hindsight vision afforded by the claimed invention; and

"(D) Reasonable expectation of success is the standard with which obviousness is determined."

In the present case, the Examiner has first of all failed to consider the claimed invention as a whole. The claimed invention has to do with static program analysis in which properties of the behavior of a program are determined without actually executing it. This is done by representing programming objects in memory as graphical objects on a display. When program properties of the programming objects change, changes are induced in the graphical properties of the visual representation of the object on the display.

Secondly, the Examiner has failed to consider the references as a whole and demonstrate that there is a suggestion of the desirability and thus the obviousness of making the combination. In the present case, Walton et al. discloses a graphic user interface design tool, while Preston discloses a data input and retrieval apparatus. These are two entirely different systems, and there is no reasonable basis for their combination.

Thirdly, the Examiner has engaged in impermissible hindsight and compounded that with mis-representations of what the references actually disclose by using incomplete quotes out of context.

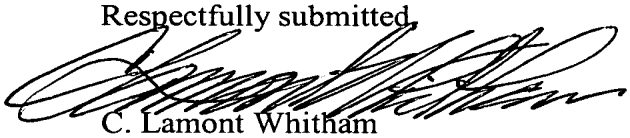
Finally, the Examiner has not shown a reasonable expectation of success. What exactly would the combination of Walton et al. and Preston produce? Would it even be an operable system? While the answers to these questions cannot be determined, what is clear is that the combination would not produce the claimed invention, and that is the standard under 35 U.S.C. §103. Under the circumstances, withdrawal of the rejection is in order and respectfully requested.

In view of the foregoing, it is respectfully requested that the application be reconsidered, that claims 1 to 9 be allowed, and that the application be passed to issue.

Should the Examiner find the application to be other than in condition for allowance, the Examiner is requested to contact the undersigned at the local telephone number listed below to discuss any other changes deemed necessary in a telephonic or personal interview.

A provisional petition is hereby made for any extension of time necessary for the continued pendency during the life of this application. Please charge any fees for such provisional petition and any deficiencies in fees and credit any overpayment of fees to Attorney's Deposit Account No. 50-0510 (IBM/Yorktown).

Respectfully submitted,



C. Lamont Whitham
Reg. No. 22,424

Whitham, Curtis & Christofferson, P.C.
11491 Sunset Hills Road, Suite 340
Reston, VA 20190
Tel. (703) 787-9400
Fax. (703) 787-7557
Customer No.: 30743